```
"""
1.  Write a Python program to reverse the
string "Programming". Print the reversed
string.
   Hint: Use string slicing or a loop.
"""
def reverse_string(s):
   return s[::-1]

string_to_reverse = "Programming"
reversed_string = reverse_string(string_to_reverse)

print(reversed_string)

"""
2.  Create a Python program that takes a
user's full name as input and prints the
initials in uppercase.
   Example: Input: "john doe", Output:
"J.D."
"""
def get_initials(full_name):
   names = full_name.split()
   initials = ""
   for name in names:
      initials += name[0].upper() + "."
```

```
    return initials[:-1]

full_name = input("Enter your full name: ")

initials = get_initials(full_name)
print(initials)

"""
3.  Write a Python program to check if a
given string is a palindrome. A palindrome
reads the same forwards
    and backward (e.g., "radar", "level").
Hint: Compare the string with its reverse.
"""
def is_palindrome(s):
    s = s.lower()
    return s == s[::-1]

string_to_check = input("Enter a string to
check for palindrome: ")
if is_palindrome(string_to_check):
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

```
"""
4.  Create a Python program that asks the
user to enter a sentence and counts the
number of words in the sentence.
    Hint: Use the split() method to break
the string into words.
"""
def count_words(sentence):
    words = sentence.split()
    return len(words)

sentence = input("Enter a sentence: ")
word_count = count_words(sentence)
print("The sentence contains", word_count,
"words.")

"""
5.  Write a Python program to replace all
occurrences of "is" with "was" in the
string "This is a string and it
    is an example." Print the modified
string.
"""
def replace_is_with_was(text):
    return text.replace("is", "was")
```

```
original_string = "This is a string and it
is an example."
modified_string =
replace_is_with_was(original_string)
print(modified_string)
```